PENDING CLAIMS AND STATUS THEREOF

1. **(currently amended)** A method of preventing processing errors due to ~~**invalid pointers**~~ **pointer registers having invalid data**, comprising:

setting trap flags associated with pointer registers to a reset condition after one of a power-up of a processor or a reset of the processor;

fetching **from a one of the pointer registers** an instruction having a pointer operand for execution;

determining whether a trap flag corresponding to the ~~**pointer**~~ **one of the pointer registers** is in a set or reset condition**, wherein the set condition indicates the one of the pointer registers has valid data and the reset condition indicates the one of the pointer registers has invalid data**; and

generating a trap control signal when the trap flag is in a reset condition.

2. (original) The method according to claim 1, further comprising:

triggering a trap interrupt based on the trap control signal.

3. **(currently amended)** The method according to claim 1, further comprising:

executing the instruction **when the trap flag is in the set condition**.

4. **(currently amended)** The method according to claim 1, further comprising:

changing the trap flag corresponding to the pointer from the reset condition to the set condition based upon a write **of valid data** to the pointer.

Claim 5 (canceled)

6. **(currently amended)** A method of preventing processing errors due to ~~**invalid**~~ pointers **having invalid data**, comprising:

providing trap flags, each corresponding to a pointer register;

~~**resetting**~~ **setting** the trap flags **to a reset condition** upon one of a power up or reset; and

setting the trap flag corresponding to each pointer register based on **valid data being written to** the pointer register ~~**being written**~~.

7. **(currently amended)** The method according to claim 6, further comprising:

generating a trap control signal when an instruction **causes the processor to read [[reads]]** a pointer register **having a corresponding [[with a]]** trap flag set to **the** reset **condition**.

8. (original) The method according to claim 7, further comprising:

triggering a trap interrupt based on the trap control signal.

9. **(currently amended)** A processor **having logic** that prevents processing errors due to ~~**invalid pointers**~~ **pointer registers having invalid data**, comprising:

instruction fetch and decode logic for fetching and decoding instructions;

registers **for holding data**;

trap flags **associated with the registers**, each **trap flag** corresponding to a **one of the registers** ~~**register**~~ and each **trap flag** indicating a set or reset condition**, wherein the trap flags are in the reset condition after a processor power-up or reset and in the set condition after valid data is written to the associated register**; and

a pointer trap coupled to the trap flags, the pointer trap generating a trap control signal based on decoding an instruction that reads a **one of the registers** ~~**register**~~ that has a corresponding trap flag in **the** **[[a]]** reset condition**, whereby the trap flag in the reset condition indicates the register has invalid data**.

10. **(currently amended)** The processor according to claim 9, wherein the trap control signal is only generated when the register being read from is **[[acting]]** **being used** as a pointer register.

11. (original) The processor according to claim 9, further comprising interrupt logic for generating an interrupt based on the trap control signal.

12. **(currently amended)** The processor according to claim 9, further comprising:

a reset/power up unit coupled to the trap flags, **wherein** the reset/power on unit ~~**resetting**~~ **causes** the trap flags **to be in the reset condition** upon a reset/power up.

13. **(currently amended)** The processor according to claim 9, further comprising:

a trap flag control unit coupled to the trap flags, **wherein** the trap flag control unit ~~**resetting**~~ **causes** the trap flags **to be in the reset condition** upon a reset/power up and **to be in the set condition** ~~**setting each trap flag based on**~~ **when** the corresponding ~~**register having been**~~ **registers have valid data** written **thereto** ~~**by an instruction**~~.